I.J. Image, Graphics and Signal Processing, **2011, 3, 1-9**
Published Online April 2011 in MECS (http://www.mecs-press.org/)

User Name Alias Extraction in Emails

Meijuan Yin, Junyong Luo, Ding Cao, Xiaonan Liu and Yongxing Tan
Information Science and Technology Institute, Zhengzhou 450002, China
Email: raindot_ymj@163.com, luojunyong@vip.371.net,
{nine_day, caoding8483}@163.com, wuming621@126.com

***Abstract*—Finding out user identity information from emails is one of the important research topics in email mining. Most approaches extract an email user's name only from the header of an email, but there are often many name information appearing in the body of emails, and those names are usually more suitable for representing the sender's or recipient's identity. This paper focuses on the problem of extracting email users' name aliases in the body of plain-text emails. After locating and extracting salutation and signature blocks from email bodies, we can identify the potential aliases in the salutation and signature lines, which can be directly associated with the corresponding email address in email headers, by using named entity recognition(NER) tools. However the identified aliases may be half-baked or there are still some potential aliases that can't be correctly identified. So we propose a novel approach to efficiently and accurately extract aliases in the salutation and signature lines based on name boundary word template built on the characteristics of alias neighboring words. Results on the public subset of the Enron corpus indicate that the approaches presented in this paper can efficiently extract user's aliases from email bodies.**

***Index Terms*—Emails, Alias Extraction, Entity resolution, Salutation and signature blocks, Name boundary word template**

I. INTRODUCTION

The popularity of the Internet makes people can communicate in many ways, such as Email, Blog, MSN and OICQ et al. To protect privacy, people usually use names different from their real-world names (denoted as aliases) when communicating with each other on the Internet in a period of time. Alias in the paper is a general appellation for formal name and informal name such as anonym, nickname, shortened form and so on. However, a user customarily employs a relatively fixed alias in one kind of communication ways. So aliases can indicate some important information about the user's identities to a certain extent, and alias extraction in emails becomes a new important research topic of email mining and identity mining. Mining user's identity information in network communication is a popular research topic of data mining. This technique can be used in many network applications, such as identity recognition, information retrieval, social network analysis and so on. This paper focuses only on email communication and studies the problem of extracting aliases of email users from email corpus.

Most approaches extract a user's name only from header fields of an email, such as "To" Header and "From" Header, but there are often many names in the body of emails, which are usually more suitable for representing the sender's or recipient's real identity. This paper focuses on the problem of extracting email users' names from the body of emails. In email bodies, aliases that appear in the salutation and signature lines can be directly associated with the corresponding email addresses in email headers. While aliases that appear in the pure content except salutation and signature lines in email bodies may refer to the email sender or recipient whose email address appearing in the email headers, and also may refer to another user except the email sender and recipient whose email address usually does not come out in the email. So to these aliases appearing in the pure content of email bodies, after extracting them from emails, then we must make sure whom the alias refers to, which is another research problem that is resolving personal name references in the full email. Consequently we only focus on aliases that appear in the salutation and signature lines in email bodies in this paper.

The first step for alias extraction from salutation and signature lines is locate and elicit salutation and signature blocks from email bodies. And as methods to locate salutation and signature block for plain-text emails are very different from that for other emails and relatively easier, so we only take plain-text emails into account in this paper. To effectively extract salutation and signature block from the body of an email, we propose the salutation and signature blocks locating algorithm based on statistical and rules restricted methods, which is presented in our former work [1].

After having locating and elicit salutation and signature lines from email bodies, we can use Named entity recognition (NER) or part-of-speech tagging tools to identify the potential aliases in the salutation and signature lines. But the identified aliases may be half-baked or there are still some potential aliases that can't be correctly identified. So we propose a novel approach to extract aliases in the salutation and signature lines. Using name boundary word template built on the characteristics of alias neighboring words, which only come with the aliases appearing salutation and signature lines, we verify and amend the potential aliases that were identified by NER tools. Results on the public subset of the Enron collection indicate that the approaches presented in this paper can efficiently extract user's aliases from the body of emails.

The remainder of this paper is organized as follows. Section II reviews earlier approaches to alias detection and email user identity modeling. The system framework to extract users' aliases from salutation and signature blocks in email bodies are introduced in section III. The algorithm to extract aliases in salutation and signature blocks proposed in this paper is introduced in detail in section IV. In section V, the method is evaluated on the public subset of the Enron collection. Results of our approach are concluded and future works are discussed in section VI.

II. Related Work

Our approach of extracting aliases in emails relates to but different with the more general problem referred as "Entity Resolution." Entity resolution is generically defined as a process of determining the mapping from references (e.g. names, phrases) observed in data to real-world entities (e.g. persons, locations). Although much has been done on entity resolution [2], extracting aliases of personal names has not received enough attention. D. Bollegala et al. [3, 4, 5, 6] exploit trained models to extract candidate aliases of a given real personal name from Web pages. The approach can extract an entity's aliases, but the extracted aliases can't be associated with the email address of the entity. We restricted our work to the email collection. And as the email body is unstructured, it is difficult to elicit aliases from email bodies via model-based methods.

Identifying aliases of an email user is important for name reference resolution and entity's identity modeling in emails. C. Bird et al. [7] study the problem of correctly relating aliases and email addresses that belong to the same entity by clustering. They extract (alias, address) pairs from the header of emails and cluster them by the similarity between the pairs. C. Diehl et al. [8] firstly explore the problem of resolving personal name references in the full email including the message body. They build email communication social network based on the email sender-recipient relationship, and resolve the personal name references by using header-based traffic analysis techniques. Neither of those two studies reported the difficult problem of extracting users' name aliases from email bodies.

T. Elsayed et al. [9, 10, 11] also address the problem of resolving personal name references in the full email including the message body. They regard the email address as the key attribute to describe an entity identity, elicit names related to the email address from the header, the salutation and signature of an email, and resolve the personal name references by building the entity identity model. The method to partition the content and signature in the body of an email by using blank lines in Elsayed's research is very simple but effect for emails with normal bodies. When the body of an email is not consistent with the standard format, the method only using blank lines does not work well. Besides, to extract aliases from the salutation and signature, they do not directly use NER tools, but remove stop words and invalid sentences via a set of simple rules, compare the remaining lines with the user name in the email address, and select a whole line with some words similar to the user name as an alias of the user name.

Since above researches suffer from precision in extracting aliases of email users from the full email message, this paper is to propose a novel method to accurately and efficiently extract email user's aliases from email bodies.

III. Alias Extraction System

A. Outline

We use an email address to stand for the corresponding user of the email address and all names related with the address are defined as the user's name aliases, which include his formal names and informal names such as anonym, nickname, shortened form and so on. Though a person may actually hold two or more email addresses simultaneously, to simplify the problem at the first research on alias extraction we ignore this case in this paper. And that to combine several email addresses related with actually the same person to one user, that is email address alias merging, is another important research problem in email mining. Furthermore, results of this paper can be used to resolve the problem of email address alias merging.

The target of our alias extraction system is extracting all name aliases of each email user from the given email corpus, to provide alias information for the future research about alias authority analysis and identity identifying. The framework of our email user's name alias extraction system is outlined in Fig.1 and is composed of three modules: Alias Extraction in Email Header (Module 1), Salutation and Signature Blocks Locating (Module 2) and Alias Extraction in Blocks (Module 3). Using a set of email corpus as input, the system outputs all aliases of each email user appearing in the email corpus in the form of (email, alias) pairs. Firstly, in Module 1, we elicit (email, alias) pairs of the sender and recipients from the address fields such as From, To, Cc and Bcc in email headers. Then, in Module 2, we locate and extract salutation and signature blocks from email bodies. At last, In Module 3, we recognize and extract candidate aliases in the extracted text blocks, then related them to the email addresses elicited from the corresponding header of the same email and generate new (email, alias) pairs. The design of our alias extraction system is described in detail in the next section.

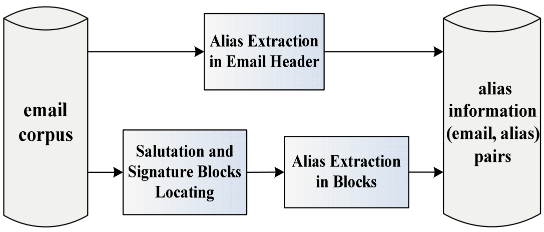

email corpus
Alias Extraction in Email Header
Salutation and Signature Blocks Locating
Alias Extraction in Blocks
alias information (email, alias) pairs

Figure 1. Framework of email user's name alias extraction system.

B. Design of Alias Extraction System

For a common email message, aliases that can be directly associated with the email address of the email sender or recipient can be found in two locations. One is email address fields such as “From”, “To”, “Cc” and “Bcc” in email headers where the alias of a user appears with his corresponding email address. The other is salutation and signature blocks in email bodies, where only the alias of a user appears. A typical Enron email message is shown in Fig.2.

In each line of address fields in the Header part, aliases of email users appear in a fixed style, that is the alias of a user is included by a pair of double quotation marks, e.g. “a name alias”, after which is the user’s email address included by a pair of brackets, e.g. <an email address>. So in the module of Alias Extraction in Email Header, we first extract aliases from “From”, “To”, “Cc” and “Bcc” address fields in Header part by matching the punctuation pairs ‘ “ ’ and ‘ ” ’, and email addresses by matching the punctuation pairs ‘ < ’ and ‘ > ’, then associate each alias with the email address next it and draw an alias of a user in the form of (email, alias) pair.

In the Body part, only aliases appearing in the lines of salutation block and signature block can be directly related to the corresponding email address extracted by Module 1. An alias extracted from the signature block of an email body should be associated with the email address elicited from the “From” header of the same email, while an alias extracted from the salutation block of a email body can only be associated with the corresponding email address in the “To” header of the same email. However, in the salutation and signature blocks of email bodies, aliases appear not in a fixed style. So to extract aliases from the Body part, we must resolve two key problems: one is the salutation and signature blocks locating and the other is aliases identifying and extracting accurately. After locating and eliciting salutation and signature blocks, we can use part-of-speech tagging tools to label block texts and identify candidate aliases. But names appear in email bodies are usually informal names, which results in that only using named entity recognizing tools the candidate aliases identified may not entire or even some aliases would be omitted.

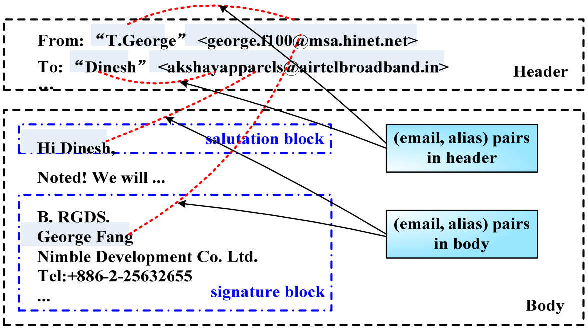

From: “T.George” <george.f100@msa.hinet.net>
To: “Dinesh” <akshayapparels@airtelbroadband.in>
...
Header
Hi Dinesh,
salutation block
Noted! We will ...
B. RGDS.
George Fang
Nimble Development Co. Ltd.
Tel:+886-2-25632655
...
signature block
Body
(email, alias) pairs in header
(email, alias) pairs in body

Figure 2. Example of a typical email message.

Consequently, we propose Salutation and Signature Blocks Locating Algorithm based on statistical and rules restriction methods (abbreviated to SSBLA), which basically resolve the first problem, and present a novel approach called Name Boundary Word Template based Alias Extraction Algorithm(abbreviated to NBWT_AEA) to extract aliases in the salutation and signature blocks, which greatly settle the second problem.

The framework and main process flow of our alias extraction system are illustrated in Fig.3. The input of the system is email corpus and the output is users’ alias information in the form of (email, alias) pairs. There are primarily four steps in the process flow of the system. The first step is directly extracting (email, alias) pairs from “From, To, Cc and Bcc” headers. The second one is locating and eliciting salutation and signature blocks in the email body via SSBLA. The third one is using NER or part-of-speech tagging tools to label names in the text of salutation and signature blocks and obtaining candidate aliases. The last step is building and exploiting name boundary word template to verify and amend candidate aliases to valid aliases via NBWT_AEA, and then associating each valid alias with the corresponding address extracted from “To” or “From” Header in the first step. In this way, we can finally extract all (email, alias) pairs and find all aliases of each user appearing in the email collection. The key algorithms of the system are described in the next section, especially algorithm NBWT_AEA is presented in detail in section IV.B and evaluated in section V.

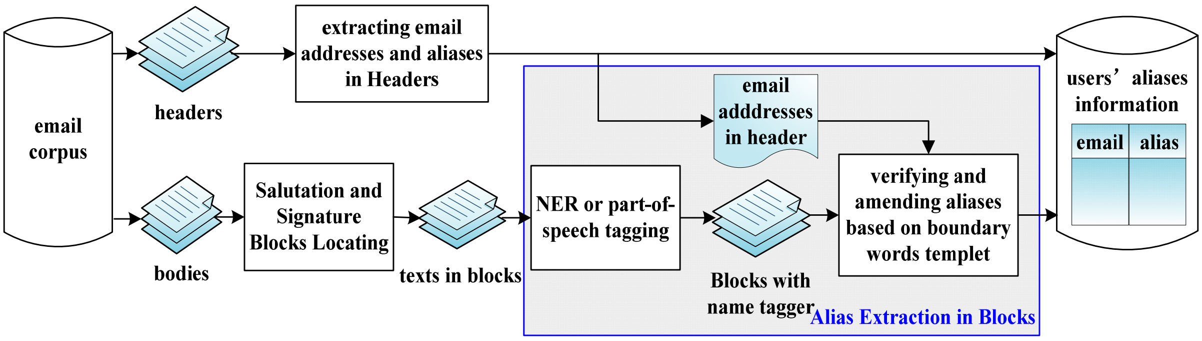

email corpus
headers
extracting email addresses and aliases in Headers
bodies
Salutation and Signature Blocks Locating
texts in blocks
NER or part-of-speech tagging
email adddresses in header
Blocks with name tagger
verifying and amending aliases based on boundary words templet
Alias Extraction in Blocks
users’ aliases information
email
alias

Figure 3. Process flow of alias extraction system.

Copyright © 2011 MECS

IV. Extracting Alias in salutation and signature blocks

In this part, we describe the method to accurately extract aliases from the salutation and signature blocks located by SSBLA. We first briefly introduce the basic idea of algorithm SSBLA, which is amply presented and evaluated in our former work [1]. And then describe the definition of Name Boundary Word Template, and present the Name Boundary Word Template based Alias Extraction Algorithm (NBWT_AEA) in detail.

A. Salutation and Signature Blocks Locating Algorithm

Algorithm SSBLA is based on statistical and rules restriction methods. The basic idea is we first exploit the statistical method to roughly estimate the number of lines in salutation and signature blocks, and then introduce some restriction rules to refine the lines located by the statistical method and elicit the lines that exactly belong to the salutation and signature blocks.

The locating algorithm can simultaneously extract salutation and signature lines. As we used both the statistical method and rules restriction method, the salutation and signature blocks locating algorithm can greatly improve the locating efficiency and promise a relatively high accuracy of the extracted blocks at the same time, when compared with other methods.

B. Definition of Name Boundary Word Templates

In the alias extraction system, after having located and elicited salutation and signature blocks from email bodies, we use part-of-speech tagging tools to label block texts and identify candidate aliases. There are some relatively mature part-of-speech tagging tools in different languages. We take emails in English and Chinese as examples in this paper.

For English emails, we choose the well-known named entity recognition tool in English nature language process field, Named Entity Recognizer System Version1.1.1 of Stanford University (abbreviated to Stanford NER) [12]. The label of names tagged by NER is a pair of labels "<PERSON>" and "</PERSON>", and between the pair of labels is a person name. For example, a result tagged by NER is "<PERSON>Jim Jarmusch</PERSON>", and the string "Jim Jarmusch" between the label pair "<PERSON>" and "</PERSON>" is an English person name. For Chinese emails, we select the famous part-of-speech tagging tool in Chinese nature language process field, ICTCLAS 3.0 of Chinese Academy of Sciences (CAS) [13]. The part-of-speech tagging label for a person name is "/nr", and it includes four sub-labels: "/nr1" is for a Chinese family name, "/nr2" is for a Chinese Christian name, "/nrj" is for a Japanese name, and "/nrf" is for a transliteration name. The Chinese characters before the label "/nr" is a potential person name.

By using above named entity recognizing tools, we can identify most of the format names in salutation and signature blocks. However, names appear in email bodies are usually informal names such as anonyms, nicknames, short names, honorific names and so on, which results in that candidate aliases identified by only using named entity recognizing tools may not entire or even some aliases would be omitted in the tagging process of named entity recognizing tools. For example, the infrequent Chinese name "Li Shuo" is labeled by ICTCLAS 3.0 as "Li/nr1 Shuo/ag", based on which we can only extract the family name "Li", and for the Chinese nickname "Little Dou" tagged as "Little/a Dou/n", we can't extract any part of the nickname. According to above analysis, identifying aliases in email bodies only by named entity recognizing tools must induce inaccurate and may miss some aliases.

It is well-known that in the text of salutation and signature blocks in emails names usually appear with some idiom before them, such as "Dear", "Hello", "Yours truly", "Sincerely" and so on, and with a space or an end symbol of the text line (e.g. the control character "CRLF") after them. So in this paper, we exploit this feature of words neighboring names to improve the accuracy of extracting aliases in salutation and signature blocks. We build a name boundary word template for general use based on the feature of words around names in email salutation and signature blocks, and then use the template to amend aliases identified by named entity recognizing tools or discover new aliases omitted by named entity recognizing tools.

The steps of building our name boundary word template are as follows.

a) Step 1: Defining the length of potential names.

Names in all kinds of languages commonly exist in two forms: one is the formal way and the other is informal, such as anonyms, nicknames, shortened names and so on. A formal name is usually composed of two parts (the family name or names and the Christian name), while an informal name may be composed of only one part (the family name or the Christian name) or two parts (the family name or the Christian name plus a salutation word to express respect, intimacy or title). So we define the length, the minimum length and the maximum length of a potential name according to the length of each part.

Definition 1: length, Min-length, Max-length of a potential name.

$l(x)$: is the length of a word sequence x, that is the number of minimum language element in the sequence, e.g. the minimum language element in Chinese is a Chinese character, in English is an English word;

There are two notes about the mathematical expression $l(x)$:

- if x is a punctuation, then $l(x)=1$; and $l(x)=0$ denote x doesn't exist;
- if n is a potential name, the length of name n can be expressed as $l(n)$.

$L_{n\min}$: is the minimum possible length of a name;

$L_{n\max}$: is the maximum possible length of a name.

For example, in general, for a Chinese name, $L_{n\max}=4$ and $L_{n\min}=1$, and for an English name, then

$L_{n\max} = 3$ and $L_{n\min} = 1$.

b) Step 2: Building front and rear boundary words list according to the special context words neighboring a name in email salutation and signature blocks.

By analyzing a large amount of email messages in Enron email corpus [13] and referring to related information about greeting words in letters, we find many front and rear boundary words of names in email salutation and signature blocks, of which the most frequent boundary words are shown in Table I.

Definition 2: length, Max-length of a front or rear boundary word.

f : is the word sequence of a front boundary word, then the length of f is $l(f)$. And if f is a punctuation or special character, e.g. SPACE、CRLF, then $l(f) = 1$;

r : is the word sequence of a rear boundary word, then the length of r is $l(r)$. And if r is a punctuation or special character, e.g. SPACE、CRLF, then $l(r) = 1$;

$L_{f\max}$: is the maximum length of any front boundary word f , in English $L_{f\max} = 2$;

$L_{r\max}$: is the maximum length of any rear boundary word r , in English $L_{r\max} = 1$.

c) Step 3: Defining name boundary word template in email salutation and signature blocks.

Definition 3: name boundary word template FNR1.

In email salutation and signature blocks, if there is a word sequence < *fnr* >, which satisfies that f is one of the front boundary words in Table I, r is one of the rear boundary words in Table I, n or part of n is a name labeled by NER tools, and $0 <= l(f) <= L_{f\max}$, $0 <= l(r) <= L_{r\max}$, $l(f) \times l(r) \neq 0$, $L_{n\min} <= l(n) <= L_{n\max}$, then named < *fnr* > as a name boundary word template in email salutation and signature blocks, and the temple is marked as FNR1.

The above template is only fit for names identified by NER tools. We must define another template for names that have not been identified by NER tools.

Definition 4: name boundary word template FNR2.

In email salutation and signature blocks, if there is a word sequence < *fnr* >, which satisfies that f is one of the front boundary words in Table I, r is one of the rear boundary words in Table I, n is an arbitrary word sequence, and $0 < l(f) <= L_{f\max}$ 、 $0 < l(r) <= L_{r\max}$ 、 $L_{n\min} <= l(n) <= L_{n\max}$, then named < *fnr* > as a name boundary word template in email salutation and signature blocks, and the temple is marked as FNR2.

There are two differences in above two templates. One is that in FNR1 n is a name tagged by NER tools while in FNR2 n is an arbitrary word sequence. The other is that in FNR1 it is not necessary that both the front and rear boundary word (f and r) exist, while in FNR2 it is necessary that both f and r exist to ensure at the farthest that the arbitrary word sequence n between f and r is an alias. That is to say, aliases extracted according to the template FNR1 are relatively more accuracy than those found by the template FNR2. So the template FNR1 is used in the case that we have found a name by using NER tools in email salutation and signature blocks and amend the name labeled by NER tools. Only when we can’t learn any name by using NER tools for email salutation and signature blocks, then the template FNR2 is used to find new possible aliases omitted by NER tools.

C. Alias Extraction Algorithm

The basic idea of Name Boundary Word Template based Alias Extraction Algorithm is: if there is a name having been identified by NER tools in email salutation and signature blocks, then directly use name boundary word template FNR1 to amend the front and rear of the name, and get the corresponding alias to be extracted; otherwise, that is to say there is no name having been identified by NER tools, employ name boundary word template FNR2 to locate the word sequence n whose front and rear boundary words can both be affirmed, and the word sequence n is the alias to be extracted.

Definition 5: related mathematical symbols and expressions in our alias extraction algorithm.

T : is the text of salutation or signature blocks in email bodies having been labeled by NER tools;

$w(i)$: is the i th minimum language element in T (e.g. the i th word in English text);

$w(i)..w(i+j)$: is a word sequence in T ;

n : is the word sequence n labeled as a personal name by NER tools;

x : is the sequence number of n in T , that is the sequence number of the first word of n in T , namely $w(x)$ is the first word of n .

a) Alias extraction sub-algorithm based on the template FNR1.

The sub-algorithm to amend a potential alias having been labeled as a name by NER tools based on name boundary word template FNR1 (abbreviated to AEA_FNR1) includes four essential steps. The first step is judging whether there is a rear boundary word listed in Table I after name n in T ; If there is indeed a rear boundary word r , then the second step is amending the rear of name n according to r ; The third step is judging whether there is a front boundary word listed in Table I after name n in T ; If there is indeed a front boundary word f , then the last step is amending the front of name n according to f .

Copyright © 2011 MECS

TABLE I.
LIST OF NAME BOUNDARY WORDS IN ENGLISH EMAIL SALUTATION AND SIGNATURE BLOCKS

Front boundary words	Dear, My dear, Hi, “Hi,”, Hello, “Hello,”, Honorable, Hon. , Yours, Yours sincerely, Sincerely yours, Sincerely, Yours faithfully, Faithfully yours, Yours truly, Truly yours, Yours respectfully, Respectfully yours, cordially……
Rear boundary words	‘,’, SPACE, CRLF……

b) Alias extraction sub-algorithm based on the template FNR1.

The sub-algorithm to amend a potential alias having been labeled as a name by NER tools based on name boundary word template FNR1 (abbreviated to AEA_FNR1) includes four essential steps. The first step is judging whether there is a rear boundary word listed in Table I after name n in T; If there is indeed a rear boundary word r, then the second step is amending the rear of name n according to r; The third step is judging whether there is a front boundary word listed in Table I after name n in T; If there is indeed a front boundary word f, then the last step is amending the front of name n according to f.

Given the list of name boundary words Table I, text T of salutation or signature blocks in email bodies having been labeled by NER tools, n labeled as a personal name by NER tools, and x the sequence number of n in T, the procedure of sub-algorithm AEA_FNR1 is shown in Fig. 4.

c) Alias extraction sub-algorithm based on the template FNR2.

The sub-algorithm to find a potential alias having been omitted by NER tools based on name boundary word template FNR2 (abbreviated to AEA_FNR2) includes three essential steps. The first step is judging whether there is a front boundary word listed in Table I in T; If there is indeed a front boundary word f, then the second step is judging whether there is a rear boundary word listed in Table I after r in T; If true, then the last step is judging the length of the word sequence between f and r, if it is not more than $L_{n\max}$, then the word sequence between f and r is a potential name.

```
Procedure AEA_FNR1(Table 1, $T$, $n$, $x$)

if ( $l(n) \geq L_{n\max}$ )   return $n$;
if ( ( $x + l(n) \leq a < l(T)$ ) && ( $0 \leq b < L_{r\max}$ ) && ( $a + b < l(T)$ ) && ( $w(a)..w(a+b)$ is a rear boundary word in Table 1) )
{
    $temp \leftarrow w(x)..w(a-1)$;
    if ( $l(temp) <= L_{n\max}$ )  $n \leftarrow temp$;
    if ( $l(n) == L_{n\max}$ )   return $n$;
}
if ( ( $0 \leq a < x$ ) && ( $0 \leq b < L_{f\max}$ ) && ( $a + b < x$ ) && ( $w(a)..w(a+b)$ is a front boundary word in Table 1) )
{
    $temp \leftarrow w(a+b+1)..w(x+l(n)-1)$;
    if ( $l(temp) <= L_{n\max}$ )  $n \leftarrow temp$;
}
return $n$;
```

Figure 4. Procedure of the sub-algorithm AEA_FNR1.

Given the list of name boundary words Table I, text T of salutation or signature blocks in email bodies having been labeled by NER tools, the procedure of sub-algorithm AEA_FNR2 is shown in Fig. 5.

d) Alias extraction algorithm based on name boundary word template.

According to above two sub-algorithms, the steps of our Name Boundary Word Template based Alias Extraction Algorithm (abbreviated to NBWT_AEA) are as follows. The first step is judging whether there is a word sequence n having been labeled as a name by NER tools in text T of salutation or signature blocks in email bodies; If true, then the second step is getting the sequence number of n in T, and the last step is calling procedure AEA_FNR1( ) to amend the front and rear of name n based on template FNR1; If false, then the second step is directly calling procedure AEA_FNR2( ) to find a potential alias n that can’t be identified as a name by NER tools based on template FNR2.

Given the list of name boundary words Table I, text T of salutation or signature blocks in email bodies having been labeled by NER tools, the procedure of algorithm NBWT_AEA is described in Fig. 6.

```
Procedure AEA_FNR2(Table 1, $T$)

$n \leftarrow NULL$;
if ( ( $0 \leq i < l(T)$ ) && ( $0 \leq a < L_{f\max}$ ) && ( $i + a < l(T)$ ) && ( $w(i)..w(i+a)$ is a front boundary word in Table 1) )
    if ( ( $i + a < j < l(T)$ ) && ( $0 \leq b < L_{r\max}$ ) && ( $j + b < l(T)$ ) && ( $w(j)..w(j+b)$ is a rear boundary word in Table 1) )
        if ( $j - (i+a) > 1$ )
        {    $temp \leftarrow w(i+a+1)..w(j-1)$;
             if ( $l(temp) <= L_{n\max}$ )  $n \leftarrow temp$;
        }
return $n$;
```

Figure 5. Procedure of the sub-algorithm AEA_FNR2.

```
Procedure NBWT_AEA(Table 1, $T$)

$n \leftarrow NULL$;
if ( ( $0 \leq x < l(T)$ ) && ( $j \geq 0$ ) && ( $x + j < l(T)$ ) && ( $w(x)..w(x+j)$ is a name) )
{
    $n \leftarrow w(x)..w(x+j)$;
    if ( $l(n) < L_{n\max}$ )  $n \leftarrow AEA\_FNR1(Table1, T, n, x)$;
}
else
    $n \leftarrow AEA\_FNR2(Table1, T)$;
return $n$;
```

Figure 6. Procedure of our alias extraction algorithm NBWT_AEA.

Copyright © 2011 MECS

The above algorithm NBWT_AEA can be used for emails in different languages. If only properly setting the list of name boundary words, the minimum language element, and the value of other related variables in above definitions, then you can apply the algorithm to extract aliases from email salutation and signature blocks in other languages. In the experiment, we only test our algorithm on English emails, and the result is very good.

V. EVALUATION

A. Dataset

In this section we analyze and validate the availability of our method to extract alias only from salutation and signature blocks of email bodies. The experiments are carried out on the public Enron collection [14] published by Federal Energy Regulatory Commission(FERC) in 2003. It contains emails sent among 150 employees of the Enron corporation from October, 1998 to June, 2002. A part of those emails include salutation and signature blocks with different kinds of format, and the experiment results of our Salutation and Signature Blocks Locating Algorithm (SSBLA) [1] on those emails have shown a relatively high performance. The emails in the collection are stored in folders, each folder correspond to one user and include several sub-folders such as sent_mail folder, inbox folder, all_documents folder and so on. In the experiments we select the sent_mail folders of 20 users, which include 6065 emails, and from those folders we randomly choose 2000 emails which include names in "email-name" lists appended to the dataset. In those 2000 emails, after removing the quoted text from the email body, only 1672 emails have the text body, in which 358 emails include salutation blocks and 971 emails include signature blocks. About 3.2% of those emails with salutation or signature blocks do not include any names, and 1287 names appear in those salutation and signature blocks by labeling manually.

B. Experiments and Evaluation

We take the text segment of above 358 salutation blocks and 971 signature blocks labeled manually as the test dataset in our experiments. First, we extract all valid aliases from the dataset by using alias extraction methods to be tested in the experiments. Then associating them with the corresponding email addresses elicit form "From" header or "To" header, and build the (email, name) pairs. At last, by comparing the pairs with the "email-name" lists and the results labeled manually, we can work out the precision and recall of each method and testify the validity of our Name Boundary Word Template based Alias Extraction Algorithm (NBWT_AEA).

We use three methods to extract aliases from salutation and signature blocks. In method 1, we directly employ Stanford NER system to tag names in those blocks and elicit those names as aliases, and then associate them with email users. In method 2, we extract aliases by using the sub-algorithm AEA_FNR1, which amend those aliases labeled and extracted in method 1 according to template FNR1. In method 3, we use our alias extraction algorithm NBWT_AEA based on both template FNR1 and template FNR2 to extract aliases, which can not only amend those aliases labeled and extracted in method 1, but also find new potential aliases omitted by method 1 and method 2.

In the evaluation step, we treat an alias of email user associated with his email address as a correct one if the alias string matches the labeled result. To evaluate the performance of the three methods, we use three measures: precision rate P, recall rate R and F-measure F, which are usually used to evaluate the performance in the Information Retrieval system. The formulas are defined as in (1).

$$P = n_{cname} \big/ n_{ename} \ ,$$

$$R = n_{cname} \big/ n_{rname} \ ,$$

$$F = 2PR/(P+R)\ . \qquad (1)$$

n_{ename} : is the number of aliases extracted from the email dataset by the alias extraction methods; n_{cname} : is the number of correct aliases in all of the extracted aliases; n_{rname} : is the total number of aliases labeled manually in the email dataset. Precision rate and recall rate reflect two different aspects of locating performance, and the F measure reflects the integrated quality of alias extraction methods. Table II shows the evaluation results of three methods in above dataset.

Table II shows that our approach NBWT_AEA to extract aliases is much better than method 1 that only use NER tools to label aliases in both the precision and recall rate.

TABLE II.
EVALUATION RESULTS OF THREE ALIAS EXTRACTION METHODS ON ABOVE DATASET

	n_{rname}	n_{ename}	n_{cname}	Precision(%)	Recall(%)	F measure(%)
method 1 (NER)	1287	698	672	96.28	52.29	67.77
method 2 (AEA_FNR1)	1287	698	686	98.28	53.30	69.12
method 3 (NBWT_AEA)	1287	1079	1053	97.59	81.82	89.01

Table II shows that our approach NBWT_AEA is much better than method 1 that only use NER tools to label aliases in both precision and recall rate.

Compared with method 1, method 2 can't extract aliases omitted by method 1 when using Stanford NER label names yet, so aliases extracted by method 2 are as many as those of method 1. But by using template FNR1 in method 2, 14 of half-baked names extracted by method 1 are amended correctly. Therefore, though the recall of method 2 grows little, the precision increases much and reaches 98%. For the other 12 aliases that extracted incorrectly by method 1, method 2 didn't extract them correctly yet. The reason is that most of them are half-baked names and don't match template FNR1, and a few of them essentially are not names but are tagged as names by Stanford NER tool.

As for method 3, the approach in dealing with aliases identified by method 1 is the same as method 2. And we find that there are many aliases in above dataset were tagged as not names but other entities due to some inherent errors of Stanford NER system, for example that Phillip and Theresa are labeled as place name, and Darrell and Shelley are labeled as organization names. For most of those names, their appearance coincides with template FNR2 built in this paper, for example that a name is at the beginning of a line and next of the name is a comma, or that before the name is "Hi," which is at the beginning of a line and next of the name is CRLF. And for a few of those names not matching template FNR2, method 3 can't identify them yet. So method 3 can easily extract those aliases by using FNR2, and the recall rate of method 3 arrives at 98%, which is much higher than that of method 1 and method 2. However, as the precision of extracting aliases based on template FNR2 is less than that based on template FNR1, the precision of method 3is little less than method 2, but higher than method 1. In conclusion as for the integrated quality, the F measure of method 3 is about 90%, which is much higher than those of the other two methods.

Besides, there are some exceptional emails, such as names appear in signature blocks do not express contact information, and few signature blocks include more than one names. These exceptions have influence to the precisions of all three methods.

VI. Conclusion and Future Works

In this work we addressed the problem of automatically extracting aliases of email users from the full email message. The limitation of most existing related works is extracting aliases only from email address headers in email headers, such as "From", "To", etc, which makes the insufficient usage of email messages and the information about aliases extracted aren't overall. In allusion to this limitation, we proposed the novel approach to extract aliases of email sender and recipient from salutation and signature blocks in email bodies. After having located and extracted salutation and signature blocks from email bodies, we used Stanford NER system for English emails to identify the potential aliases in the salutation and signature lines, which can be directly related with the email addresses in email headers. To amend aliases that were identified by using NER tools and find new potential aliases omitted by NER tools at the farthest, we defined the name boundary word templates built on the characteristics of alias neighboring words, and thus obtained more valid and intact aliases. Results on the public subset of the Enron corpus indicate that the alias extraction method presented in this paper can efficiently extract users' aliases from email bodies.

As name boundary word templates defined in this paper are based on the feature of words around names in email salutation and signature blocks of emails, our alias extraction algorithm NBWT_AEA based on name boundary word template is fit for extracting aliases only in salutation and signature blocks of email bodies, but can't be used for aliases appearing in other part of email bodies. Besides, our approach can amend most of half-baked names labeled by NER tools according to template FNR1, but can't find and delete pseudo aliases that are not names in nature but tagged as names by NER tools. Future studies on alias extraction in emails will investigate novel methods that can be applied to the full email and can identify pseudo aliases. Since our method is fit for emails in any language after properly setting the list of name boundary words and values of variables in related definitions, we will do more experiments on emails in other language, such as Chinese emails. By using our method we can usually extract more than one alias for only one user. How to rank the authority of each alias of one user to get the most authoritative alias that can be used to represent the user's identity is another future work.

References

[1] M. Yin, J. Luo, D. Cao, X. Liu and M. Li. Automatically locating salutation and signature blocks in emails[A]. Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'11) [C]. Shanghai, China, 2011, in press.

[2] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In The SIAM International Conference on Data Mining (SIAM-SDM),Bethesda, MD, USA, 2006.

[3] D. Bollegala, Y. Matsuo, and M. Ishizuka. Disambiguating personal names on the web using automatically extracted key phrases. In Proc. of the 17th European Conference on Artificial Intelligence, pages 553-557, 2006.

[4] D. Bollegara, Y. Matsuo, and M. Ishizuka. Extracting key phrases to disambiguate personal names on the web. In Proc. CICLing 2006, 2006.

[5] D. Bollegala, T. Honma. Identification of Personal Name Aliases on the Web[A]. In: Proceedings of WWW 2008 Workshop on Social Web Search and Mining(SWSM 2008). Beijing, China, 2008.

[6] D. Bollegala, T. Honma, Y. Matsuo, and M. Ishizuka. Mining for personal name aliases on the web. In: Proceeding of the 17th international conference on World Wide Web, April 21-25, 2008, Beijing, China.

[7] C. Bird, A. Gourley and A. Swaminathan. Mining Email Social Networks[A]. In: Proceedings of the 2006

international workshop on Mining software repositories [C]. Shanghai, China, 2006: 137-143.

[8] Chris Diehl, Lise Getoor, and Galileo Namata. Name reference resolution in organizational email archives. In Proceddings of SIAM International Conference on Data Mining,Bethesda, MD , USA, April 20-22 2006.

[9] T. Elsayed, Oard D W. Modeling Identity in Archival Collections of Email[A]. In: Proceedings of the Third Conference on Email and Anti-Spam[C]. Mountain View, California, USA, 2006.

[10] T. Elsayed, D. W. Oard, and G. Namata. Resolving personal names in email using context expansion. In Association for Computational Linguistics(ACL), 2008.

[11] T. Elsayed, G. Namata, L. Getoor, and D. W. Oard. Personal name resolution in email: A heuristic approach. Technical Report UMIACS LAMP-TR-150, University of Maryland, March 2008.

[12] Stanford University. Named Entity Recognition System [EB/OL]. http://nlp.stanford.edu/software/stanford-ner-2009-01-16.tgz. 2009.

[13] Chinese Academy of Sciences(CAS). Institute of Computing Technology Chinese Lexical Analysis System(ICTCLAS) version 3.0 [EB/OL]. http://www.ictclas.org/down/ictclas2009/window_c_32.rar. 2009.

[14] The email collection of Enron Corporation [DB/OL]. http://www.cs.cmu.edu/~enron/. 2003.

Meijuan Yin was born in Anhui Province, China at November, 1977. She was conferred a M.Sc. in computer science by the university of Zhengzhou Information Science and Technology Institute at Zhengzhou, China, in 2003. She is working on the Ph.D. in computer software and academic of the same University.

After graduating from the University of Zhengzhou Information Science and Technology Institute at Zhengzhou, China, she became an assistant of the University in 2003 and turned to a lecturer in 2005. Her current research interests include data mining, social network analysis, and information security.

Ms. Yin joined China Computer Federation (CCF) as a common member in 2007 and received the IEEE membership in 2010.

Junyong Luo received a M.Sc. in computer science and engineering from the university of Zhengzhou Information Science and Technology Institute at Zhengzhou, China, and became a teacher of the university in 1992.

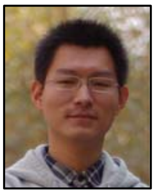
He was developed into a professor and doctoral supervisor of computer science and engineering in 2005. His research has covered many areas, including database, network security, data mining, and information security. His current research projects are on knowledge discovering, social network analysis and parallel computing.

Ding Cao was born in Henan province, China. She holds a bachelor's degree in computer science from Zhengzhou Information Science and technology Institute at Zhengzhou, China in 2007.

She is working on master's degree in computer science and her research interests include data mining and file type identification.

Xiaonan Liu was born in Liaoning Province, China. He was conferred a M.Sc. in computer science by the University of Zhengzhou Information Science and Technology Institute at Zhengzhou, China, in 2006. He is working on the Ph.D. in computer software and academic of the same University.

He graduated from the University of Zhengzhou Information Science and Technology Institute at Zhengzhou, China, and became an assistant of the University in 2000 and turned to a lecturer in 2006. His research interests include binary translation, compile, and decompile.



Yongxing Tan, born in Henan province, China, in 1989, holds a bachelor's degree in computer science from Zhengzhou Information Science and technology Institute at Zhengzhou, China in 2009. He is a second-year graduate student for the Master degree of engineering in the same university.

His main research interests include the fields of information extraction and machine learning.